OBSERVACIONES DE LA PRACTICA

Estudiante 1 Felipe Carvajal Cod 202014203

Estudiante 2 Valentina Perea Márquez Cod 202013095

# **Ambientes de pruebas**

|  |  |  |
| --- | --- | --- |
|  | Máquina 1 | Máquina 2 |
| Procesadores | AMD Ryzen 7 5800H with Radeon Graphics 3.20 GHz | AMD Ryzen 5 3400G with Radeon Vega Graphics 3.70 |
| Memoria RAM (GB) | 24,0 GB | 16,0 GB |
| Sistema Operativo | Windows 10 64 bits | Windows 10 Pro 2004 64 bits |

Tabla 1. Especificaciones de las máquinas para ejecutar las pruebas de rendimiento.

# **Maquina 1**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 10% | 13418 | 101737,38 | 546,88 | EX.RECURSION | 359,38 |
| 80.00% | 5000 | 418437,5 | 580,70 | 345.23 | 181.25 |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 10% | 13418 | Más de 10 minutos | Mas de 10 minutos | +10 MIN | 31390,63 |
| 80.00% | 5000 | Más de 10 minutos | 2468,47 | 0.0 | 1584.34 |

Tabla 3. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 101737,38 | Más de 10 minutos |
| *Shell Sort* | 546,88 | Mas de 10 minutos |
| *Merge Sort* | 359,38 | 31390,63 |
| *Quick Sort* | +10 MIN | +10 MIN |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

# **Maquina 2**

## **Resultados**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (ARRAYLIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 10% | 13418 | 134640.62 | 234.37 | EX.RECURSION | 421.87 |
| 80.00% | 5000 | 18890.625 | 234.375 | 453.125 | 171.875 |

Tabla 2. Comparación de tiempos de ejecución para los ordenamientos en la representación arreglo.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Porcentaje de la muestra [pct]** | **Tamaño de la muestra (LINKED\_LIST)** | **Insertion Sort [ms]** | **Shell Sort [ms]** | **Quick Sort [ms]** | **Merge Sort [ms]** |
| 10% | 13418 | +10 MIN | 618265.62 | +10 MIN | 35828.12 |
| 80.00% | 5000 | +5 MIN | 66640.62 | 0.0 | 5015.625 |

Tabla 3. Comparación de tiempos de ejecución para los ordenamientos en la representación lista enlazada.

|  |  |  |
| --- | --- | --- |
| Algoritmo | Arreglo (ARRAYLIST) | Lista enlazada (LINKED\_LIST) |
| *Insertion Sort* | 18890.625 | +5 MIN |
| *Shell Sort* | 234.37 | 66640.62 |
| *Merge Sort* | 171.875 | 5015.625 |
| *Quick Sort* | 453.125 | 0.0 |

Tabla 4. Comparación de eficiencia de acuerdo con los algoritmos de ordenamientos y estructuras de datos utilizadas.

# **Preguntas de análisis**

1. ¿El comportamiento de los algoritmos es acorde a lo enunciado teóricamente?

Es dificil explicar solo con una tabla del tiempo el comportamiento de los algoritmos, sin embargo, evidenciamos cierta similaridad con la teoria.

INSERTION(O(n)): Este compara los datos adyacentes, de todos los procesos fue el mas lento, y obtenia tiempos muy altos. Peor caso (O(n\*\*2))

SHELLSORT (O(n log (n))): El que menos demoro.

MERGESORT (O(n log (n))): Mientras mayor sea el numero de elementos, mayor es el numero de subdivisiones que tiene que hacer por lo que vemos un aumento en el tiempo a medida que la cantidad de datos tambien crece, pues se demora en llegar al final, es decir, si se comporto como en la teoria.

QUICKSORT (O(n log (n))): La complejidad es similar a la de un Mergesort, pero tiene un peor desempeno cuando los datos crecen, se demora menos con datos pequenos y a medida que crecen el tiempo se va elevando.

1. ¿Existe alguna diferencia entre los resultados obtenidos al ejecutar las pruebas en diferentes máquinas?

Si, aunque la maquina 2 no estaba tan alejada del procesador de la maquina 1, la memoria conto con un rol importante al comparar y ordenar los datos, la diferencia aveces es mayor a 50 ms en la maquina 1.

1. De existir diferencias, ¿a qué creen que se deben?

Quiza la diferencia radica en la cantidad de almacenamiento ocupado en la memoria del equipo, ya que al estar muy llena tiene a actuar con más lentitud, ademas de la memoria Ram que le permite matener mas los datos temporales e incluso el procesador de nuevas generaciones que es mas veloz.

1. ¿Cuál Estructura de Datos funciona mejor si solo se tiene en cuenta los tiempos de ejecución de los algoritmos?

La estructura de datos que mejor funciona es Array\_List, ya que todas las pruebas efectuadas con este tipo de lista tomaron significativamente mucho menos tiempo.

1. Teniendo en cuenta las pruebas de tiempo de ejecución por todos los algoritmos de ordenamiento estudiados (iterativos y recursivos), proponga un ranking de los mismo de mayor eficiencia a menor eficiencia en tiempo para ordenar la mayor cantidad de obras de arte.

* 1. Shell ()
* 2. Merge ()
* 3. Insertion ()
* 4. Quicksort ()